BOOSTED MMI FOR MODEL AND FEATURE-SPACE DISCRIMINATIVE TRAINING

Daniel Povey, Dimitri Kanevsky, Brian Kingsbury,
Bhuvana Ramabhadran, George Saon and Karthik Visweswariah

IBM T.J. Watson Research Center
Yorktown Heights, NY, USA
{dpovey,kanevsky,bedk,bhuvana,gsaon,kv1}@us.ibm.com

ABSTRACT

We present a modified form of the Maximum Mutual Information (MMI) objective function which gives improved results for discriminative training. The modification consists of boosting the likelihoods of paths in the denominator lattice that have a higher phone error relative to the correct transcript, by using the same phone accuracy function that is used in Minimum Phone Error (MPE) training. We combine this with another improvement to our implementation of the Extended Baum-Welch update equations for MMI, namely the canceling of any shared part of the numerator and denominator statistics on each frame (a procedure that is already done in MPE). This change affects the Gaussian-specific learning rate. We also investigate another modification whereby we replace I-smoothing to the ML estimate with I-smoothing to the previous iteration's value. Boosted MMI gives better results than MPE in both model and feature-space discriminative training, although not consistently.

Index Terms— MMI, MPE, Maximum Margin, Discriminative Training, Speech Recognition

1. INTRODUCTION

There has recently been some interest in large margin techniques for speech recognition. In the large margin approach described in [1, 2], a margin is enforced which is proportional to the Hamming distance between the hypothesized utterance and the correct utterance - i.e. the number of frames for which the HMM state which aligns to that frame is different from the one from the correct transcript. We apply a similar idea to the MMI objective function by boosting the likelihood of hypothesized utterances proportional to their difference from the correct utterance. We also introduce improvements to the optimization prodedure.

In Section 2 we introduce the Maximum Mutual Information criterion and explain how we optimize it for discriminative training; in Section 3 we introduce the boosted MMI objective function; in Sections 4 and 5 we describe our improvements to the update equations and review feature space discriminative training. Sections 6 and 7 give experimental results and conclusions.

2. MAXIMUM MUTUAL INFORMATION

The Maximum Mutual Information (MMI) objective function [3, 4, 5] seeks to maximize the posterior probability of the correct utterance given our models:

$$\mathcal{F}_{\mathrm{MMIE}}(\lambda) = \sum_{r=1}^{R} \log \frac{p_\lambda(\mathcal{X}_r|\mathcal{M}_{s_r})^\kappa P(s_r)}{\sum_s p_\lambda(\mathcal{X}_r|\mathcal{M}_s)^\kappa P(s)}, \tag{1}$$

where λ represents the acoustic model parameters, $\mathcal{X}_r$ are the training utterances, $\mathcal{M}_s$ is the HMM sequence corresponding to a sentence s, and s_r is the correct transcription for the r'th utterance, κ is the acoustic scale as used in decoding and $P(s)$ is a weakened language model such as a unigram.

Our optimization of the MMI objective function uses the Extended Baum-Welch update equations and it requires accumulating two sets of the normal kind of statistics used for E-M updates of HMMs, via forward-backward accumulation on two HMMs for each utterance. The numerator HMM (corresponding to the numerator in Equation 1) is the correct transcription, and the denominator HMM is a recognition model containing all possible words. Statistics accumulation is done by generating lattices once and doing forward-backward on the lattices on each iteration. This not only reduces computation but allows us to more correctly optimize Equation 1 because it allows us to apply the acoustic weight κ at the word level; application at the state level does not lead to as good results [4]. Forward-backward statistics accumulation on the lattices gives us two sets of statistics which we distinguish with the superscripts num and den. EBW is an iterative procedure which we apply for about four iterations for best results.

The Gaussian means and variances are updated as follows (valid for full covariances):

$$\hat{\boldsymbol{\mu}}_{jm} = \frac{\boldsymbol{x}^{\mathrm{num}}_{jm} - \boldsymbol{x}^{\mathrm{den}}_{jm} + D_{jm}\boldsymbol{\mu}_{jm}}{\gamma^{\mathrm{num}}_{jm} - \gamma^{\mathrm{den}}_{jm} + D_{jm}} \tag{2}$$

$$\hat{\boldsymbol{\Sigma}}_{jm} = \frac{\boldsymbol{S}^{\mathrm{num}}_{jm} - \boldsymbol{S}^{\mathrm{den}}_{jm} + D_{jm}(\boldsymbol{\Sigma}_{jm} + \boldsymbol{\mu}_{jm}\boldsymbol{\mu}^T_{jm})}{\gamma^{\mathrm{num}}_{jm} - \gamma^{\mathrm{den}}_{jm} + D_{jm}} - \hat{\boldsymbol{\mu}}_{jm}\hat{\boldsymbol{\mu}}^T_{jm} \tag{3}$$

where j and m are the HMM-state and Gaussian index respectively, γ_{jm} is the "count" of the Gaussian, and $\boldsymbol{x}_{jm}$ and $\boldsymbol{S}_{jm}$ are the standard weighted sums over features $\boldsymbol{x}(t)$ and $\boldsymbol{x}(t)\boldsymbol{x}(t)^T$ respectively. For the diagonal-covariance case the above would be done with just the diagonal elements of the variance. The Gaussian-specific learning-rate constants D_{jm} are set to the largest of (i) the typical case: $E\gamma^{\mathrm{den}}_{jm}$, where E is a constant typically set to 2; and (ii) double the smallest D_{jm} that would lead to $\hat{\boldsymbol{\Sigma}}_{jm}$ being positive definite. We can easily handle case (ii) by trying the update using a D_{jm} equal to half of that in case (i), then increasing it if necessary by small steps

This work was funded by DARPA contract HR0011-06-2-0001

until $\hat{\mathbf{\Sigma}}_{jm}$ is positive definite (or all positive in the diagonal case), and then doubling it to get the final value.

For a performance improvement we can also do I-smoothing [4], which amounts to gradually backing off to the ML estimate as the counts get small. We can represent I-smoothing in the most general way as follows as a modification to the statistics:

$$\begin{aligned} \boldsymbol{x}_{jm}^{\mathrm{num}} &:= \boldsymbol{x}_{jm}^{\mathrm{num}} + \tau\boldsymbol{\mu}^{\mathrm{b}} && (4) \\ \boldsymbol{S}_{jm}^{\mathrm{num}} &:= \boldsymbol{S}_{jm}^{\mathrm{num}} + \tau(\boldsymbol{\mu}^{\mathrm{b}}{\boldsymbol{\mu}^{\mathrm{b}}}^{T} + \mathbf{\Sigma}^{\mathrm{b}}) && (5) \\ \gamma_{jm}^{\mathrm{num}} &:= \gamma_{jm}^{\mathrm{num}} + \tau, && (6) \end{aligned}$$

where $\boldsymbol{\mu}^{\mathrm{b}}$ and $\mathbf{\Sigma}^{\mathrm{b}}$ are the parameter values we are backing off to, and τ (e.g. $\tau = 100$ for MMI, or $\tau = 50$ for MPE) is a constant we introduce. In the normal case of backing off to the ML estimate, $\boldsymbol{\mu}^{\mathrm{b}}$ and $\mathbf{\Sigma}^{\mathrm{b}}$ are estimated from the ML statistics on the current iteration which in the MMI case are the same as the numerator statistics.

The mixture weights c_{jm} are optimized as follows [4] (recent experiments show that this gives roughly a 0.5% *relative* improvement versus leaving them unchanged). For a particular HMM state j we optimize an auxiliary function given by

$$\sum_{m=1}^{M} \gamma_{jm}^{\mathrm{num}} \log c_{jm} - \frac{\gamma_{jm}^{\mathrm{den}}}{c'_{jm}} c_{jm}, \qquad (7)$$

where c'_{jm} is the previous value. This is done iteratively by setting $c_{jm}^{(0)} = c'_{jm}$ and for (e.g.) $p = 0 \ldots 100$, doing $c_{jm}^{(p+1)} = \frac{\gamma_{jm}^{\mathrm{num}} + k_{jm} c_{jm}^{(p)}}{\sum_m \gamma_{jm}^{\mathrm{num}} + k_{jm} c_{jm}^{(p)}}$, where $k_{jm} = \left(\max_m \frac{\gamma_{jm}^{\mathrm{den}}}{c'_{jm}}\right) - \frac{\gamma_{jm}^{\mathrm{den}}}{c'_{jm}}$. Transition probabilities (not used by us) can be updated like mixture weights.

2.1. Minimum Phone Error

The MPE objective function, which has some relevance to boosted MMI, is the sum of the phone accuracies of all possible sentences given the reference, weighted by their likelihood as a function of the model:

$$\mathcal{F}_{MPE}(\lambda) = \sum_{r=1}^{R} \frac{\sum_s p_\lambda(\mathcal{X}_r|\mathcal{M}_s)^\kappa P(s) A(s, s_r)}{\sum_s p_\lambda(\mathcal{X}_r|\mathcal{M}_s)^\kappa P(s)}, \qquad (8)$$

where $A(s, s_r)$ is the raw phone accuracy of s given the reference s_r, which equals the number of correct phones minus the number of insertions. This raw accuracy must be approximated for efficiency [6]. We also recently investigated [7] replacing $A(s, s_r)$ with various other functions including a state-level accuracy which we call state-level Minimum Bayes Risk (s-MBR) after [8]. This works as well as or better than the phone-level accuracy, and is the same as the Hamming distance used in the large margin techniques in [1, 2].

3. BOOSTED MMI

In boosted MMI the objective function is:

$$\mathcal{F}_{\mathrm{MMIE}}(\lambda) = \sum_{r=1}^{R} \log \frac{p_\lambda(\mathcal{X}_r|\mathcal{M}_{s_r})^\kappa P(s_r)}{\sum_s p_\lambda(\mathcal{X}_r|\mathcal{M}_s)^\kappa P(s) \exp(-bA(s, s_r))}, \qquad (9)$$

where b (e.g. $b = 0.5$) is a boosting factor which we introduce. We boost the likelihood of the sentences that have more errors, thus generating more confusable data. Boosted MMI can viewed as trying to enforce a soft margin that is proportional to the number of errors in a hypothesised sentence, as in [1, 2]. Note that $A(s, s_r)$, which is the accuracy of a sentence s given the reference s_r, can be computed in various ways and this must be specified along with b. Unless otherwise stated we compute it per phone, the same way as for normal MPE. The extra computation involved in BMMI as opposed to MMI is very small: while we are doing the forward-backward algorithm on the denominator lattice, for each arc in the lattice we subtract from the acoustic log-likelihood b times the contribution to the sentence-level accuracy $A(s, s_r)$ arising from the arc in question. This behaves like a modification to the language model contribution on each arc.

4. IMPROVEMENTS TO MODEL-SPACE UPDATES

4.1. Canceled statistics

The first improvement we made to the optimization procedure is to cancel the statistics accumulated on each frame from the numerator and denominator. This is something that happens anyway in our implementation of MPE. It means that if the Gaussian-specific occupancy on a particular time t is nonzero for both numerator and denominator models we cancel any shared part, so

$$\gamma_{jm}(t)^{\mathrm{numc}} := \gamma_{jm}(t)^{\mathrm{num}} - \min(\gamma_{jm}(t)^{\mathrm{num}}, \gamma_{jm}(t)^{\mathrm{den}}) \qquad (10)$$
$$\gamma_{jm}(t)^{\mathrm{denc}} := \gamma_{jm}(t)^{\mathrm{den}} - \min(\gamma_{jm}(t)^{\mathrm{num}}, \gamma_{jm}(t)^{\mathrm{den}}). \qquad (11)$$

We are using the superscripts numc and denc for the canceled numerator and denominator statistics; we omit the utterance subscript r for convenience. The only effect this canceling has is to change the Gaussian-specific learning-rate constant D_{jm}. Note that if we are doing the normal form of I-smoothing to the ML estimate, we must store the unmodified numerator (num) statistics.

4.2. I-smoothing to previous iteration

The other change which we introduce here is I-smoothing to the previous iteration. In I-smoothing for MMI we have previously backed off to the ML estimate. As mentioned in [9], in our current implementation of MPE we back off to the MMI estimate (based on one iteration of EBW starting from the current iteration's statistics). We show here that I-smoothing to the previous iteration can be better than I-smoothing to the ML estimate. This is simpler to implement as it no longer involves changing the numerator statistics; we can simply change our rule for obtaining D_{jm} to be the largest of: (i) the typical case: $\tau + E\gamma_{jm}^{\mathrm{den}}$, or (ii) double the smallest D_{jm} that would lead to $\hat{\Sigma}_{jm}$ being positive definite.

5. FEATURE-SPACE DISCRIMINATIVE TRAINING

We have previously introduced feature-space MPE (fMPE) [10], with important features of our current implementation described in [11]. Feature-space discriminative training gives a large part of our improvement from discriminative training. Here we introduce fMMI, which is the application of the same techniques to the MMI objective function (we use "fBMMI" if boosted MMI is the objective function). Only a single modification to our recipe is necessary other than the changed objective function. As mentioned in [11], we set the learning rate in fMPE based on a target objective function improvement of 0.04 or 0.06 on the first iteration (0.04 for tasks with low WER or where the amount of training data is very large and hence the criterion improvement due to overtraining is expected to be small. This is reduced for fMMI to 0.01 or 0.015 (0.01 for experiments reported here), to compensate for the lower range of the MMI objective function; this leads to similar E values of around 6 to 12; in fact, it would probably be better to simply set the E used in fMPE to a fixed value such as 7 for both MPE and MMI-based training.

6. EXPERIMENTAL RESULTS

We report results here on a large number of different experimental conditions; we use an identifier for each condition and the details of each are shown in Table 1. All systems have quinphone cross-word context; all have cepstral mean subtraction and some have cepstral variance normalization (not shown in table); any VTLN and/or fMLLR adaptation is indicated. All setups with fMLLR are SAT trained unless otherwise indicated. Setups that are shown as having fMLLR are also tested with regression tree MLLR following the fMLLR.

Identifier (#states,#Gauss)	Description
ABN2300 (5k, 400k)	2300h Arabic broadcast news (GALE program), VTLN+fMLLR. Test: Eval06
ACTS80 (3.5k, 75k)	80h Iraqi Arabic conversational telephone speech (LDC2006S45, GALE), VTLN+fMLLR. Test: LDC-devtest (2.5h)
EBN50 (2.2k, 50k)	50h English broadcast news, unadapted. Test: RT-04.
EBN430 (6k, 300k)	430h English broadcast news, VTLN+fMLLR, no SAT. Test: RT-04.
EBN700 (6k, 250k)	700h English broadcast news, unadapted. Test: RT-04.
ECTS175 (4.2k, 150k)	175h English conversational telephone speech, VTLN+fMLLR. Test: internal test set.
ECTS2000 (8k, 200k)	2000h English conversational telephone speech, VTLN+fMLLR. Test: internal test set.
EPS80 (8k, 190k)	175h English EU parliamentary speech (TC-STAR project), VTLN+fMLLR. Test: dev06+eval06

Table 1. Experimental conditions

6.1. Model-space BMMI vs MPE

The following experiments give results for model-space training. Both boosting and canceling give substantial improvements starting from the MMI baseline, combining to make the MMI-based results slightly better than MPE.

Objf (I-smoothed→ Objf)	WER
ML	25.3
MPE→ MMI	21.6
MPE → ML	22.3
MMI→ ML	23.0
MMI-c→ ML,$b = 0.0$	22.4
BMMI-c→ ML,$b = 0.25$	22.0
BMMI-c→ ML,$b = 0.5$	21.7
BMMI-c→ ML,$b = 0.75$	21.6
BMMI-c→ ML,$b = 1.0$	21.5
BMMI-c→ ML,$b = 2.0$	22.1
MPE→BMMI-c→ML, $b = 0.5$	24.2

Table 2. EBN50: MPE vs (B)MMI, 4th iter, κ=0.053

Table 2 gives results for model-space training in the EBN50 setup. The baseline is MPE I-smoothed to MMI (MPE → MMI). We denote the use of canceled statistics with the suffix -c (e.g. BMMI-c). X→Y means the criterion X smoothed to Y with I-smoothing; for MPE→ X we always use $\tau = 50$, and for MMI→X we use $\tau = 100$. As we can see, unmodified MMI is worse than MPE but BMMI with canceled statistics (-c) is the same or slightly better with the appropriate value of b. We get 0.6% of improvement from canceling statistics and an additional 0.9% from boosting. However MPE smoothing to BMMI-c does not work, showing a large degradation on iteration 4. This probably relates to training too fast.

Objf (smoothed→ objf)	WER	
ML	20.5	
	Iteration	
	2	4
	κ=0.1	
MPE→ MMI	18.8	18.1
MPE→ ML	18.9	18.6
MMI→ ML	19.5	18.9
BMMI→ ML,$b = 0.5$	19.3	18.6
BMMI-c→ ML,$b = 0.5$	18.4	17.9
	κ=0.053	
BMMI-c→ ML,$b = 0.5$	17.8	17.3
MPE→BMMI-c→ ML,$b = 0.5$	17.6	20.1

Table 3. EBN700 setup: MPE vs (B)MMI

Table 3 shows similar experiments on another English Broadcast News setup with 700 hours of training data. Most experiments are with κ=0.1 but we confirm the improvement of changing the acoustic weight to 0.053 with our best experiment (BMMI-c→ ML,$b = 0.5$). Again smoothing from MPE to BMMI-c leads to overtraining. The overall conclusion is again that BMMI-c→ ML is better than MPE→ MMI. With these experiments we can how much of the improvement versus plain MMI due to the boosting versus canceling: 0.3% comes from boosting and 0.7% from the canceling of statistics.

6.2. I-smoothing

Objf (smoothed→ objf)	WER (RT-04)	
ML	25.3	
	Iteration	
	2	4
MPE→ MMI	22.6	21.6
MPE→ ML	22.9	22.3
MPE→ prev	22.8	22.1
BMMI-c→ ML,$b = 0.5$	22.3	21.7
BMMI-c→ prev,$b = 0.5$	22.1	21.3
BMMI-c $b = 0.5$	22.0	22.5

Table 4. EBN50: I-smoothing→ ML vs → previous iteration (κ=0.053).

We also investigated I-smoothing to the previous iteration rather than the ML estimate. This appears to give an improvement for model-space only training, but after feature space training it does not appear to help. Table 4 shows the effect of I-smoothing to the ML estimate versus the previous iteration, for both MPE (τ=50) and MMI (τ=100). Smoothing to the previous iteration is better, by 0.2% for MPE and 0.4% for BMMI-c. In other experiments, we compared smoothing to the previous iteration versus ML for BMMI-c,$b = 0.5$ after boosted fMMI discriminative training (b=0.5) for two setups: EBN50 and EBN430. In both cases the WER was unchanged (for absolute values, see Table 5).

6.3. Feature space BMMI experiments

Setup	κ	ML	fMPE	+MPE	fBMMI	+BMMI-c
ABN2300	0.1	32.8	29.0			
	0.053				29.7	
ACTS80	0.1	43.2	41.1	38.4		
	0.053	43.2	40.4	36.8	37.5	35.9
EBN50	0.1	25.3	21.4	20.5		
	0.053	25.3	20.7	19.6	19.2	18.1
EBN430	0.053	16.2		14.0		13.1
EBN700	0.1	20.5	16.7	16.0		
	0.053	20.5	17.2		16.2	15.3
ECTS175	0.1	31.8	29.6	28.9		
	0.053	31.8	29.4	28.6	29.1	28.3
ECTS2000	0.1	27.5	24.6			
	0.053				24.3	
EPS80	0.1	8.8	7.6	7.2		
	0.053	8.8	7.6	7.2	7.3	6.8

Table 5. MPE vs BMMI for feature-space discriminative training

Table 5 compares an MPE-based and a BMMI-based recipe for feature-space followed by model-space discriminative training, on all of our training setups. BMMI is better in all but the ABN2300 setup. We do three or four iterations of feature-space discriminative training followed by two or three iterations of model-space discriminative training, except for the ABN2300 setup where we show only two iterations of feature-space training (the number of iterations is not shown in the table but in all cases we compare the same iteration). All BMMI experiments use $b = 0.5$. For model space training, in the MPE-based setup we use the backoff scheme MPE→ MMI, and in the BMMI-based setup we use MMI→ ML. The table gives results for two acoustic weights: κ=0.1 which was previously our default value, and $\kappa = 0.053$ which appears to give better results for MPE in all cases tested except EBN700 (which has the most training data of those tested, probably not coincidentally). The reader should compare the fMMI+BMMI-c number with the best fMPE+MPE number. We obtain improvements in all cases except the ABN2300 setup, where we get a degradation of 0.7%. The results suggest that an MMI based criterion may be particularly preferable when the amount of training data (per parameter) is small, something also suggested by [4]. Note also that the MPE-based results with different acoustic weights suggest that a less aggressive weighting (i.e., 0.1) may be better when the amount of training data is very large.

6.4. Optimal boosting factor

We also investigated the optimal boosting factor b for BMMI based discriminative training. The reader can refer to results given above (Table 2) for the optimal boosting factor for model-space training using BMMI-c, which appears to be around 1.0.

For feature-space BMMI, the optimal boosting factor seems to be lower and the amount of improvement appears to be smaller.

In the EBN50 setup we investigated the optimal boosting factor while attempting to minimize interactions with the learning rate by setting the E in fMMI (for training our most important transformation, see [7]) directly to 7.0. In this case the fBMMI WER after two iterations with b=(0.0,0.5,1.0,1.5) were (21.0%, 21.0%, 21.1%, 21.4%) which fails to show any improvement at all from boosting. In the EBN700 and EPS80 setups however, with the usual method of setting E we got 0.1% and 0.2% absolute improvements from changing $b = 0.0$ to $b = 0.5$, so we do get a very small improvement from boosting in the fMMI case. These results are consistent with the idea that boosting helps primarily by creating more confusable data; we believe that the amount of data is less of an issue for feature-space training as we observe that we get more WER improvement for less objective function improvement indicating less overtraining.

We also investigated generating lattices for BMMI training that reflect the same boosting of less correct utterances, using the EBN50 setup, with $A(s, s_r)$ defined as the Hamming distance between state sequences (and ten times smaller b). Our experiments failed to show any improvement from matched lattice generation.

7. CONCLUSIONS

We have introduced a modification to the MMI objective function – error-boosting – and a modification to the MMI training procedure – the canceling of statistics – that, taken together, appear to make MMI-based model-space training better than MPE. The application of error-boosting to feature-space MMI discriminative training also leads to a small improvement. We have experimented with a totally MMI-based feature and model-space discriminative training procedure and found that it sometimes but not always leads to substantial improvements over our previous MPE-based procedure.

8. REFERENCES

[1] Fei Sha and Lawrence K. Saul, “Large margin Gaussian mixture modeling for phonetic classification and recognition,” in *ICASSP*, 2006.

[2] Fei Sha and Lawrence K. Saul, “Comparison of large margin training to other discriminative methods for phonetic recognition by hidden Markov models,” in *ICASSP*, 2007.

[3] Bahl L.R., Brown P.F, de Souza P.V., and Mercer L.R., “Maximum Mutual Information Estimation of Hidden Markov Model Parameters for Speech Recognition,” in *ICASSP*, 1986.

[4] Povey D., *Discriminative Training for Large Voculabulary Speech Recognition*, Ph.D. thesis, Cambridge University, 2004.

[5] Povey D. and Woodland P.C., “Improved Discriminative Training Techniques for Large Vocabulary Speech Recognition,” in *ICASSP*, 2001.

[6] Povey D. and Woodland P.C., “Minimum Phone Error and I-smoothing for Improved Discriminative Training,” in *ICASSP*, 2002.

[7] Povey D. and Kingsbury B., “Evaluation of Proposed Modifications to MPE for Large Scale Discriminative Training,” in *ICASSP*, 2007.

[8] Gibson M. and Hain T., “Hypothesis Spaces For Minimum Bayes Risk Training In Large Vocabulary Speech Recognition,” in *Interspeech*, 2006.

[9] Hagen Soltau, Brian Kingsbury, Lidia Mangu, Daniel Povey, George Saon, and Geoffrey Zweig, “The IBM 2004 Conversational Telephony System for Rich Transcription,” in *ICASSP*, 2005.

[10] Povey D., Kingsbury B., Mangu L., Saon G., Soltau H., and Zweig G., “fMPE: Discriminatively trained features for speech recognition,” in *ICASSP*, 2005.

[11] Povey D, “Improvements to fMPE for discriminative training of features,” in *Interspeech*, 2005.